"Express Mail" mailing label number:

EL675711155US

METHOD AND APPARATUS FOR BIT ERROR RATE DETECTION

Michael H. Perrott

BACKGROUND OF THE INVENTION

Field of the Invention

The present invention relates to data communication and more particularly to evaluation of error rates associated with the data communication.

Description of the Related Art

Communication systems frequently transmit data in which the clock is embedded in the data stream rather than sent as a separate signal. When the data stream is received, a clock and data recovery circuit recovers the embedded clock and retimes the received data to the recovered clock. Traditionally, a phase-locked loop (PLL) has been used to perform the clock recovery operation. Fig. 1 shows a block diagram of a traditional PLL configured for a clock and data recovery application. The phase-locked loop 100 includes a phase detector 102, which receives the input data signal conveyed on node 104 and also receives the VCO output clock signal conveyed on node 106 from the voltage controlled oscillator (VCO) 108. The phase detector 102 generates an error signal 110, which is a function of the phase difference between the input data signal and the VCO output clock signal. The phase detector 102 may also include additional circuitry to generate the reconstructed data on output node 114.

In order to help the VCO acquire the frequency of the input data stream, it has been common to use a reference clock to center the VCO output frequency for a nominal output that approximates the frequency of the input data stream. In a typical application, the VCO will multiply the reference clock by a predetermined (or selectable factor), e.g., 16, to achieve the nominal VCO output. For example, if the multiplication factor is 16, for a 2.7 Gbps data rate, the reference clock is 168.75 MHz. The requirement for a reference clock (generally differential) adds both cost

and design complexity to the system in which a clock and data recovery circuit resides. The clock has to be supplied by a relatively high cost crystal oscillator component and distributed to the clock and data recovery circuit using design practices appropriate for high speed clock signals.

In addition to using the reference clock to center the nominal output of the PLL, the reference clock is also used to determine whether lock has been achieved. Typical lock-detect circuitry compares the reference clock to a divided down version of the recovered clock, and if the difference between the two clocks is sufficiently high, the PLL is determined to be out-of-lock.

Once the PLL is locked to the data stream, it would be desirable to know the bit error rate level. If available, bit error rate levels can help evaluate system performance and alert the system to possible degradation over time. Further, it would be desirable to be able to determine not only when lock is achieved, but also when lock is lost without the need for a reference clock.

SUMMARY OF THE INVENTION

Accordingly, a method and apparatus is provided for determining the bit error rate (BER) of an input data stream. In one embodiment, a method for determining a bit error rate in an input data stream received on an integrated circuit includes determining over a plurality of first time intervals whether at least one transition of the input data stream occurred in a predetermined phase zone of a sample clock used to sample the input data stream. A count is generated according to how many of the first time intervals have at least one transition that occurred in the predetermined phase zone. The count corresponds to the bit error rate. The correspondence may be, e.g., a correlation between the count and the bit error rate.

In another embodiment an integrated circuit for receiving an input data stream, includes a bit error detect circuit coupled to determine if a bit error occurs in the input data stream according to whether an input data stream transition occurs in a predetermined phase zone of a sample clock used in the bit error detect circuit. A counter circuit is coupled to the bit error detect circuit and supplies an indication of how many evaluation intervals have at least one bit error.

In another embodiment, the transition detect logic may also be used to infer a lock or loss of lock condition for a phase-locked loop used to recover the timing and data of the input data stream. In particular, the embodiment provides an integrated circuit for receiving an input data stream. The integrated circuit includes a phase zone detect circuit coupled to determine if a transition of the input data stream occurs in a predetermined phase zone of a sample clock used in the phase zone detect circuit. A counter circuit is coupled to the phase zone detect circuit and supplies a count indicating how many of a predetermined number of evaluation intervals have at least one transition that occurs in the predetermined phase zone. The count is used to infer whether the phase-locked loop remains locked to the timing of the input data stream.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention may be better understood, and its numerous objects, features, and advantages made apparent to those skilled in the art by referencing the accompanying drawings.

Fig. 1 shows a block diagram of a traditional PLL configured for clock and data recovery operations.

Fig. 2 shows a clock and data recovery architecture that can be modified for use with the present invention.

Fig. 3 shows additional details of the clock and data recovery circuit shown in Fig. 2.

Fig. 4A illustrates an exemplary voltage controlled oscillator (VCO) used in the present invention.

Fig. 4B is a circuit diagram of an illustrative embodiment for one of the banks of capacitors utilized in the VCO shown in Fig. 4A.

Fig. 5 depicts a common phase detector.

Fig. 6 shows a timing diagram illustrating operation of the phase detector shown in Fig. 5.

Fig. 7 illustrates the ideal placement of data transitions within the phase window of the clock used to sample the data.

Fig. 8 illustrates the movement of the transitions clockwise with respect to the 180 degree point of the sample clock in the phase detector.

Fig. 9 illustrates the effect of phase offset and increased jitter amplitude on the location of the data transitions.

Fig. 10 shows the movement of the data transitions from quadrant to quadrant as a result of frequency offset.

Fig. 11 illustrates the "forbidden zone" of the sample clock.

Fig. 12A illustrates an augmented phase detector circuit that detects transitions in the forbidden zone.

Fig. 12B illustrates another embodiment of an augmented phase detector circuit that detects transitions in the forbidden zone.

Fig. 13A illustrates data transitions at 180 degrees with respect to the phase of the sample clock signal.

Fig 13B illustrates the operation of the phase detector shown in Fig. 12A when data transitions occur in the forbidden zone.

Fig. 14 illustrates a Bernoulli process.

Fig. 15 illustrates uniform distribution of transitions across the phase range.

Fig. 16 illustrates an exemplary high level flow diagram of the control structure utilized to acquire a frequency.

Fig. 17 illustrates one preferred approach for the stepping algorithm used to vary the capacitor settings.

Fig. 18 illustrates one embodiment of an augmented phase detector that determines when transitions occur in the forbidden zone.

Fig. 19 illustrates one embodiment of an augmented phase detector that determines when transitions occur in the forbidden zone.

Fig. 20 illustrates the implementation of the delay associated with the OC3 clock and the generation of the clocks for OC24, OC12 and OC3 mode of operations.

Fig. 21 illustrates implementation of the delay associated with OC24 and OC12 clocks.

Fig. 22 illustrates interpolating between two clocks to achieve a shorter delay than can be achieved by utilizing one delay buffer.

Fig. 23 shows a block diagram of a portion of an exemplary clock and data recovery circuit incorporating the augmented phase detector.

Fig. 24A shows the probability of falsely asserting out-of-lock with various bit error rates (BER) and for various bit error counts for 16 trials.

Fig. 24B shows the probability lock is falsely declared versus count and width of the forbidden zone for 16 trials.

Fig. 25A shows the probability of falsely asserting out-of-lock with various bit error rates (BER) and for various bit error counts for 48 trials.

Fig. 25B shows the probability lock is falsely declared versus count and width of the forbidden zone for 48 trials.

Fig. 26A shows the probability of falsely asserting out-of-lock with various bit error rates (BER) and for various bit error counts for 512 trials.

Fig. 26B shows the probability lock is falsely declared versus count and width of the forbidden zone for 512 trials.

Fig. 27A shows a table that indicates one embodiment for assigning digital values indicating a bit error rate.

Fig. 27B shows a table that indicates one embodiment for assigning digital values indicating a bit error rate to achieve greater resolution.

Fig. 28 shows the probability that the measured count of transitions in the forbidden zone (the bit error count) is greater than the bit error rate for 512 trials when the PLL is in-lock.

Figs. 29-31 illustrate the operation of stepping the capacitance values of the VCO while trying to acquire lock.

Figs. 32-40 illustrate simulation of frequency detection under various conditions.

Fig. 41 shows a block diagram of a clock and data recovery integrated circuit incorporating various aspects described herein.

DESCRIPTION OF THE PREFERRED EMBODIMENT(S)

Referring to Fig. 2, a clock and data recovery architecture 200 that can be utilized in conjunction with the present invention is illustrated. As seen in Fig. 2, a phase detector 210, an augmented version of which is described further herein, is followed by a delta-sigma modulator 212. The phase detector 210 compares the phase of the input data signal conveyed on node 214 to the phase of the recovered clock signal conveyed on node 216, and generates a pulse width modulated error signal waveform. In this example, the error signal is a current waveform flowing into or out of node 218. The delta-sigma modulator 212 then converts the pulse width modulated error signal into a discrete-time and discrete-amplitude digital output signal, in this example generating a one-bit digital output on its output node 220.

As shown in greater detail in Fig. 3 in one embodiment, the delta-sigma modulator 212 includes a modest-sized (e.g., having a typical value of 2-3 pF) integrating capacitor 222 connected to node 218, and further includes a digital comparator block 226 which samples the voltage on its input node 218 when clocked by a delta-sigma clock received on clock node 224. The digital output generated on the output node 220 is fed back as a negative current by feedback block 228 into node 218 to provide the requisite feedback into the integrating capacitor 222 of the delta-sigma modulator. Operation of such first-order delta-sigma modulators is well-known to one skilled in the art. Further details on the clock and data recovery circuit shown

in Fig. 2 can be found in application number 60/217,208, filed July 10, 2000, which is incorporated herein by reference in its entirety.

Referring again to Fig. 2, the loop filter for this clock and data recovery circuit includes a feedforward path 230 formed by a gain block 232 and a filter block 234, and further includes a digital integrating path 240. The output signal of the feedforward path 230, which is conveyed on node 236, and the output signal of the digital integrating path 240, which is conveyed on node 249, are combined by summer block 250 to generate a control signal (V_g) on node 252 for the voltage controlled oscillator 260.

The digital integrating block 240 includes a decimator 242, an accumulator 244, a D/A converter 246, and a filter block 248. The digital accumulator 244, which includes a multiple-bit register to represent the cumulative value of the phase error, receives a decimated digital phase error representation from decimator 242, and increases or decreases the cumulative value accordingly. For example, if the digital phase error representation corresponds to a "leading" phase relationship, the digital accumulator 244 will increase (or alternately, decrease) the cumulative value stored in its output register. Conversely, if the digital phase error representation corresponds to a "lagging" phase relationship, the digital accumulator 244 will decrease (or alternately, increase) its cumulative value stored in its output register. While this and other block diagrams are described using the terminology of a single node connecting the blocks, it should be appreciated that, when required by the context in the various embodiments, such a "node" may actually represent a pair of nodes for conveying a differential signal, or may represent multiple separate wires (e.g., a bus) for carrying several related signals.

As described above, the multiple-bit output register of the digital accumulator 244 holds a digital representation of an integrated value of the phase error. This digital value is preferably communicated on an N-bit wide output bus 245 to the D/A converter 246 which converts the digital representation back into an analog signal. Filter block 248 provides a smoothing function to the reconstructed analog signal. The output of the filter block 248 is then conveyed on node 249 to the summer block 250.

As stated above, because a phase detector having a digital output signal is used, the requirement for a separate analog-to-digital converter, which exists only in the digital integrating block, is eliminated. This allows a low offset to be achieved in the phase error through the feedforward path because both the feedforward path and the integrating path receive the same digital signal.

An exemplary voltage controlled oscillator (VCO) 260 is shown in Fig. 4A. In one embodiment, the voltage controlled oscillator is implemented as a tank circuit (an LC oscillator) in which the frequency of oscillation depends on the inductance (L) 402 and capacitance (C) 404. In the illustrated embodiment, the output of the VCO 260 is adjusted by varying the capacitance 404, which includes separately controllable capacitor circuits. The summer circuit supplies the control voltage (V_g) on node 252 to adjust the variable capacitance 410 circuit. The voltage V_g controlling the capacitance setting is an analog voltage and thus the capacitance is indicated as being an analog capacitance. In addition to the variable capacitance circuit 410, variable capacitance circuit 412 accounts for drift and other long term variations that can affect operation of the VCO. The fixed capacitance 414 is used to provide a capacitance that approximates the desired value and can represent parasitic capacitance in the circuit. Variable capacitance circuit 416 provides a digitally controlled variable capacitance that can be used, as described further herein, to acquire the frequency of the input data stream without the use of a reference clock. Variable capacitance 416 in one preferred embodiment includes variable capacitors configured to allow coarse grained, medium grained and fine grained adjustment of capacitance settings as described further herein. An exemplary variable capacitance circuit is shown in Fig. 4B.

FIG. 4B is a circuit diagram of an illustrative embodiment for one of the banks of capacitors 430 utilized as part of variable capacitance 416. Fig. 4B includes a fixed capacitor 450 that represents parasitic capacitance plus any desired fixed capacitance. Discrete capacitance variations are achieved through a plurality of selectively activated capacitor and transistor pairs. Fig. 4B shows a first of these capacitor/transistor pairs, as capacitor 452 connected between ground 442 and the signal line 440 through the drain and source terminals of an NMOS transistor 462. NMOS transistor 462 acts as a switch to add in or leave out the capacitor 462 in the

overall capacitance of the discretely variable capacitance 430. The "on" or "off" state of NMOS transistor 462 is controlled by a first bit (B_0) of a digital control word supplied from control logic as described further herein. Similarly, additional capacitors 454, 456, and 458 are connected to ground 442 through NMOS transistors 464, 466, and 468, respectively. The NMOS transistors 464, 466 and 468 are controlled by bits $B_1 \ldots B_{N-1}$, B_N, of a digital control word selecting the capacitance setting.

For the circuit depicted in FIG. 4B with simple capacitor/switch circuits connected together in parallel, the total capacitance for the discretely variable capacitance 430 is equal to the sum of all of the capacitors having their respective switches in the "on" state. Note that the capacitors may all have the same value, providing for linear capacitance stepping or the capacitors may be of different values providing non-linear capacitance stepping. In the latter case, turning on one transistor will have a different effect than turning on another transistor. Numerous weighting schemes are possible, and the implemented weighting scheme depends upon the particular design considerations involved. Additional details on the capacitor/switch circuits and the VCO can be found in U.S. Pat. No. 6,137,372, entitled "Method and Apparatus for Providing Coarse and Fine Tuning Control for Synthesizing High-Frequency Signals for Wireless Communications", which is incorporated herein by reference in its entirety.

In an exemplary embodiment, the variable capacitance 416 includes multiple banks of capacitors 430 to provide coarse, medium and fine-grained control over the capacitance. The coarse bank of capacitors includes seven capacitor/switch circuits each controlled by one bit of a seven bit coarse grained digital control word. A medium bank of capacitors includes eight capacitor/switch circuits each controlled by one bit of an eight bit medium-grained control word. Finally, a fine bank of capacitors is provided with 6 capacitors/switch pairs controlled by a six bit digital control word. The coarse, medium, and fine banks of capacitors allow control over total capacitance with increasing granularity. In an exemplary embodiment, the capacitors in the coarse bank are 300 femtoFarads each, in the medium bank, 38 femtoFarads and in the fine-grained bank, 9 femtoFarads. The use of the variable

capacitance 416 to acquire frequency of an input data stream without the use of a reference clock is described further herein.

Before a description of the use of the architecture shown in Fig. 2 in referenceless frequency acquisition is described in more detail, a description of the operation of the phase detector 210 is provided to better understand use and operation of the augmented phase detector described herein.

Fig. 5 depicts a common phase detector that is generally well known in the art. In this circuit, a register 510 samples the input data signal conveyed on node 512 when clocked by the sample clock signal conveyed on node 514. The sample clock signal conveyed on node 514 is the recovered clock from the input data signal. XOR gate 520 generates on its output node B a variable-width pulse of duration generally equal to the time by which an input data signal transition leads the corresponding transition on node A, the output of register 510, which is controlled, of course, by the recovered clock conveyed on node 514. The delay block 530 is included to compensate for the clock-to-Q delay of register 510. When the data clock is correctly aligned to the input data signal (i.e., data clock transitions at precisely the mid-point of the data bit-intervals), the register 510 generates on its output node A, a signal that replicates the input data signal, but delayed by one-half period of the data clock, and the pulse on node B is of a duration exactly equal to one-half period of the data clock.

The latch 540 generates on its output node 542 a signal which replicates its input signal delayed by one-half period of the recovered clock. As a result, the second XOR gate 522 generates on its output node C a pulse with a duration that is equal to one-half of the period of the recovered clock. The fixed-duration pulse signal conveyed on node C is subtracted from the variable-width pulse signal conveyed on node B by summing block 550 to generate a phase error signal used to adjust the recovered clock.

Operation of the phase detector shown in Fig. 5 is illustrated by the timing diagram shown in Fig. 6. When the data clock is correctly aligned, as illustrated in Fig. 6, both pulse signals have equal duration, and the summing block 550 generates a zero-valued net error current. If the input data transition arrives too early, the pulse signal on node B is longer than the pulse signal on node C, and a net error current is

generated by the summing block 550. The remainder of the PLL is arranged to respond to the error current to adjust the phase of the data clock (i.e., the VCO output).

While Fig. 5 shows single-ended logic blocks and signals, in practice such circuits are typically implemented using fully differential circuitry, which provides enhanced noise immunity, better speed, and more consistent delays which are independent of data state. Moreover, many of the circuit blocks, such as the summing block 550, are more easily implemented and achieve better matching of currents when implemented differentially, thereby resulting in lower offsets.

As described previously, one method of determining if the VCO is correctly locked to the clock embedded in the input data stream is to compare the recovered clock to a reference clock. If a reference clock is unavailable, another approach has to be used both for initially setting the VCO frequency and for determining if lock has been achieved. As described further herein, frequency detection can be accomplished by detecting whether transitions fall into a predetermined phase zone (also referred to herein as a "forbidden zone") of the data clock used in the phase detector. The "forbidden zone" is a predetermined portion of the period of the sample clock used in the phase detector. When the PLL is in-lock, transitions typically will not fall into this zone. When the PLL is out-of-lock, transitions will often fall into the zone. By performing a test over many trials to determine whether data transitions in the input data stream are falling into the forbidden zone, it can be statistically inferred whether the PLL is in-lock.

Referring again to Figs. 5 and 6, the first register 510 of the phase detector 500 samples the data at 0 degrees, i.e., the leading edge of the recovered clock used to sample the data. Thus, data transitions of the input data stream ideally occur around 180 degrees with respect to the recovered clock to maximize the likelihood that a transition is detected given the existence of jitter and phase offset. Fig. 7 illustrates the preferred placement of data transitions 701 assuming low jitter. Fig. 8 illustrates the influence of phase offset on the placement of data transitions with respect to the phase of the recovered clock being used to sample the data in the first register of the phase detector 500 in Fig. 5. As can be seen in Fig. 8, the transitions have moved clockwise with respect to the recovered clock. Fig. 9 illustrates the effect of phase

offset and increased jitter amplitude. The data transitions can be seen to be more distributed in the phase of the recovered clock. If the jitter is sufficiently high, it can lead to bit errors such as transition 703.

If the PLL is trying to acquire the frequency of a clock embedded in an input data stream and no reference clock is available, the effect of frequency offset is one approach that can be used to determine whether the frequency is too high or too low and adjust the sample clock frequency accordingly. Frequency offset is the difference in frequency between the input data stream and the sample clock, and causes the data transitions in the input data stream to travel from quadrant to quadrant as shown in Fig. 10. The speed at which the transitions travel from quadrant to quadrant increases as the frequency offset increases. The direction of travel (clockwise or counterclockwise) indicates whether the frequency error is positive or negative. The number of transitions that occur in a quadrant before the transitions enter a new quadrant is a function of the magnitude of the frequency offset and the data transition density (assuming, e.g., a non-return to zero (NRZ) encoding).

Thus, one method of frequency detection is to determine the frequency offset by determining the order of the quadrants that the data transitions go through. That order indicates the direction of travel of the transitions and thus can be used to alter the VCO output positively or negatively according to the sign of the frequency error. However, that approach has several disadvantages. One problem is that the range of frequency detection is limited. The range of frequency detection is in part a function of the data transition density. A low transition density increases the difficulty in resolving the direction of travel of the transitions. Additionally, the range of frequency detection is limited because it is a function of quadrant sizes. A larger frequency offset can cause larger intervals between transitions in the quadrants, which can also increase the difficulty in resolving transition direction. In addition, quadrature phases are required for the largest frequency detection range, which can be difficult to implement with an LC oscillator.

In view of those limitations, another approach can be used to detect the frequency of an input data stream that overcomes those problems. Referring to Fig. 11, a small phase window or "forbidden zone" 111 can be defined. Any transitions, such as transition 112, that occur in that phase window 111, are assumed to be bit

errors. The phase window 111 (also shown as $\Delta\phi$) should be placed on the side of the zero degree mark that is farthest from the placement of the data transitions that occur due to phase offset. There may be a systemic offset due to, e.g., circuit delays, as well as random offset. The systemic offset in one embodiment rotates the placement of the data transitions clockwise from the 180 degree mark. Thus, the forbidden zone in such an embodiment is offset in a clockwise direction from the zero degree mark as shown in Fig. 11. In other embodiments, the phase zone can reside on the other side of the zero degree point.

If the PLL has not locked to the frequency of the input data stream, data transitions will eventually fall into the forbidden zone. That information can be used to determine whether the PLL has locked, and if it has not, adjust the output of the VCO until it does. The "forbidden zone" approach requires a circuit that detects transitions that occur in the forbidden zone. Fig. 12A illustrates a simplified augmented phase detector circuit 1200 that detects transitions in the forbidden zone. Comparing the phase detectors in Fig. 12A and Fig. 5, it can be seen that the top half 1201 of the phase detector circuit 1200 is substantially identical to the phase detector 500 shown in Fig. 5 and operates conventionally to generate an error signal from summer 550 indicating the phase error between the data in signal 512 and the clock signal 514 supplied from the VCO. In addition to the conventional phase detector 1201 illustrated, the phase detector of Fig. 12A further includes forbidden zone circuitry 1202 that detects transitions occurring in the forbidden zone of the clock signal 514. That circuitry includes a delay element 1203 that functions to delay the clock 514 by an amount corresponding to the size of the phase window. In an exemplary embodiment, the delay element 1203 delays the clock by 50-60 picoseconds. The amount of delay depends on such factors as the size of the phase window desired, the bit rate of the input data stream, and the bit error rate of the input data stream. The delayed clock 1204 is supplied to registers 1205 and 1206 and to the latch 1207. The XOR gate 1209 generates a pulse that indicates the difference between the retimed data on node 542 and the delayed data supplied from latch 1207. Note that the latches 540 and 1207 are configured to pass data when the clock is low.

Fig. 12B illustrates an alternative embodiment in which the clock for the register 510 is delayed rather than the clock for register 1205. Note also that rather than delaying the clock in either path, the data in either path may be delayed.

The operation of the circuit of Fig. 12A is illustrated in the timing diagrams 13A and 13B. In Fig. 13A the data is transitioning in the ideal location at 180 degrees with respect to the phase of the clock signal 514 (shown as CLK in Fig. 13A and 13B). The output from the registers and latches are shown to create a pulse on node F from XOR gate 1209. However, the duration of that pulse is short and therefore is not clocked into register 1206 by the rising edge of the delayed clock (DEL CLK). Thus, no error is detected and the SR latch 1210 is not set.

Referring to Fig. 13B, the operation of phase detector 1200 is illustrated for the case where data transitions do occur in the forbidden zone. When that occurs, the transition on Data In occurs after the rising edge of CLK conveyed on node 514 and thus is not clocked into register 510. The delayed clock 1204 (DEL CLK in Fig. 13B) is delayed sufficiently to catch the transition and thus the transition is stored in register 1206. As the transition or lack thereof propagates through the latches 1207 and 540, XOR gate 1209 generates a pulse output on node F that is clocked into register 1206 on the rising edge of the delayed clock. That in turn causes the SR latch to be set. The SR latch remains set indicating the existence of a bit error until control logic, described further herein, causes the SR latch to be reset so it can be used to detect another bit error.

As described more fully herein, the ability to detect the forbidden zone transitions allows the VCO output to be adjusted until transitions in the forbidden zone occur at a rate below the allowable bit error rate. The forbidden zone approach requires only a small delay element, e.g., utilizing a buffer, rather than needing to detect in which quadrature of the VCO output clock phase a transition occurs. In addition, the frequency capture range is not limited by those limitations associated with quadrature detection. The quadrature detection approach is "soft" in terms of evaluating whether the PLL is in-lock. The VCO is simply nudged in a certain direction when data shifts through quadrant boundaries. Thus, with a large frequency offset, the quadrature detection method may drift. In contrast, the forbidden zone approach described herein makes "hard" decisions. A frequency offset is evaluated

and then a VCO setting is either rejected or accepted. One cannot drift through the same settings as in the quadrature approach, until all VCO settings have been tried.

The probability of entering the forbidden zone in out-of-lock conditions can be made approximately the same for all frequency offsets. That facilitates a clock and data recovery circuit that can operate at a wide range of frequencies. In addition, the detection circuitry and control logic can be mostly built in digital logic allowing for easy implementation and low gate count. That can be particularly advantageous in mixed signal technologies. Note that in the forbidden zone approach, high bit error rates can result in the inability to lock. That will become more obvious as the approach is described in more detail. In contrast, the traditional quadrature approach may be fairly insensitive to bit error rates.

One way to understand the forbidden zone approach is to understand the statistical behavior associated therewith. That statistical behavior can be explained using Bernoulli processes. Referring to Fig. 14, a Bernoulli process is a series of independent Bernoulli trials, where the outcome can be a one or a zero. The probability that $x_i = 1$ is defined as p and the probability that $x_i = 0$ is therefore $1-p$. Define k as $k = \sum_{i=1}^{n} x_i$. The probability that $k = k_0$ is:

$$p_k(k_o) = \left(\frac{n!}{(n-k_o)!k_o!}\right) p^{k_o} (1-p)^{n-k_o}$$

A better numerical implementation of that equation is:

$$\frac{n!}{(n-k_o)!k_o!} = \prod_{i=1}^{k} \frac{n-k+i}{i}$$

If a probability of occurrence of an event is p, then the expected value of the first occurrence is:

$$E(l_1) = 1/p$$

With that background, assume that $x_i = 1$ if any transitions fall into the forbidden zone and that $x_i = 0$ if no transitions fall into the forbidden zone. A measurement period T is chosen over which to measure whether any transitions fall into the forbidden zone. For example, T may be 1024 times the period of the rate

data. The measurements over period T are repeated over n intervals of time. The number of intervals k is counted in which at least one transition fell into the forbidden zone:

$$k = \sum_{i=1}^{n} x_i$$

That value k is then compared to a critical count value k_c, and if $k > k_c$, the PLL is declared out-of-lock. If $k < k_c$, then the PLL is declared in-lock.

Because the PLL is being declared in-lock or out-of-lock based on detection of transitions in the forbidden zone, there exists the possibility of falsely declaring the PLL being out-of-lock because of a high bit error rate (BER) causing transitions to occur in the forbidden zone. There also exists the possibility of falsely declaring lock because sufficient transitions did not fall into the forbidden zone over a period of time. Remember that in a preferred embodiment, the data is encoded in an NRZ format or an equivalent and therefore transition density can vary based on data patterns. The probability of falsely declaring the PLL to be in-lock or out-of-lock should be sufficiently small as to be tolerable in the system in which the forbidden zone detection approach is employed. The specific equations for such probabilities are provided herein.

Note that the size of the forbidden zone can vary as a percentage of the entire period, and that a larger size has the effect of lower jitter tolerance since a larger number of data transitions caused by jitter will be determined to be bit errors.

Assume that transitions occur in the forbidden zone according to the bit error rate (BER). Calculate $P(x_i = 0) = (1\text{-BER})^m$, where m is the number of bits in the measurement period, or m = (data rate)(T).

Of course, $P(x_i = 1) = 1 - P(x_i = 0)$.

Assume that transitions are uniformly distributed across the phase range. That is illustrated in Fig. 15. That is roughly true if the period of frequency offset < period of measurement. If that is true, the probability that a transition does not occur in the forbidden zone over a period T is given by,

$$P(x_i = 0) = \left(\frac{2\pi - \Delta\Phi}{2\pi}\right)^m,$$

where m is the number of transitions in the measurement period and m = (data rate)(T)(transition density). Of course, $P(x_i = 1) = 1 - P(x_i = 0)$.

The desire to achieve a uniform transition distribution when the clock and data recovery (CDR) circuit is out-of-lock leads to the period of frequency offset being less than (or equal to) the period of measurement (evaluation interval). In an exemplary embodiment suitable for meeting SONET specifications, the minimum period of frequency offset is given by the frequency lock-in range of the CDR. For a typical SONET transfer specification, the frequency lock-in range is on the order of 1000 parts per million (ppm). Thus, the minimum offset frequency is approximately ($2.5 \text{ GHz} \times 1000 \times 10^{-6}$) = 2.5 MHz. That is, the CDR should be able to acquire a frequency that is within 2.5 MHz of the 2.5 GHz data rate. In such an application, the measurement period T = 1/(2.5 MHz) = 0.4 microseconds.

The probability that $x_i = 1$ (incorrect decision) when PLL is actually in-lock is given by,

$$P(x_i = 1) = 1 - (1 - \text{BER})^m,$$

where m is the number of bits in the measurement period, that is m = (data rate)T.

The probability that that $x_i = 1$ (correct decision) when PLL is not in-lock is given by,

$$P(x_i = 1) = 1 - \left(\frac{2\pi - \Delta\Phi}{2\pi}\right)^m,$$

where m is the number of transitions in the measurement period,

m = (data rate)T(transition density)

The probability of falsely asserting out-of-lock when PLL is in-lock is given by

$$p(k > k_c) = \sum_{i=k_c+1}^{n} \left(\frac{n!}{(n-i)!i!} \right) p^i (1-p)^{n-i}$$

where

$$p = 1 - 1(1-\mathrm{BER})^{(\text{data rate})\,\mathrm{T}}$$

The probability of falsely asserting lock when the PLL is out-of-lock is given by,

$$p(k <= k_c) = \sum_{i=n-k_c}^{n} \left(\frac{n!}{(n-i)!i!} \right) p^i (1-p)^{n-i},$$

where

$$p = \left(\frac{2\pi - \Delta\Phi}{2\pi} \right)$$

Exemplary curves for various bit error rates and various sizes of the forbidden zone will be provided after embodiments of the augmented phase detector circuit and the control circuitry used to adjust the capacitance of the LC oscillator, used for the variable oscillator in one embodiment of the invention, are examined in greater detail.

Referring now to Fig. 16, an exemplary high level flow diagram is shown that illustrates an embodiment of the control structure utilized to acquire a frequency. The state machine of the exemplary clock and data recovery circuit (CDR) begins in state 0, which is entered as a result of a power on (or other) reset or because of assertion of a lock to reference (LTR) signal. The lock to reference signal causes the CDR to lock to a reference clock (if available) or to stored capacitance values of the VCO as described further herein. In state 0, the loss of lock (LOL) signal is asserted indicating that the clock and data recovery circuit has not yet locked. When neither reset nor LTR is being asserted, the CDR enters state 1 and begins the task of acquiring the input frequency.

In state 1, an impedance setting is chosen. Prior to, or on entering state 1, several initialization steps may be taken. For example, the fine capacitor setting may be zeroed out to allow the fine capacitor setting to be swept for each medium/coarse setting. In addition, the digitally controlled capacitors may be set at their midrange to

prepare for frequency acquisition. Other details of the control logic are described further herein.

Referring now to Fig. 17, one preferred approach to the stepping algorithm used to vary the capacitor settings is illustrated. As shown in Fig. 17, the capacitor setting is varied gradually above and below a center capacitance value. With each step of the algorithm, the capacitance is varied with increasing amplitude above and then below the initial setting. That has the advantages of minimal frequency deviation after first lock is achieved. In addition, a fast acquisition is achieved when the desired setting is close to a current setting. Further, the probability of locking onto a harmonic of the desired frequency is reduced. In addition, that approach avoids large frequency deviations, which may be undesirable from a customer's viewpoint. In one embodiment, the approach of varying the capacitance above and below a center value is applied only to the coarse and medium capacitance settings. The fine capacitance settings are swept in a linear fashion for each medium/coarse setting. In one embodiment, the control logic can step through 336 different capacitance settings, including 6 fine settings, 7 course settings, and 8 medium settings. Note that the use of the stepping algorithm shown in Fig. 17 is exemplary only. Other approaches, including a completely linear approach, may also be used. In addition, the use of fine, medium and coarse capacitance settings is exemplary. Additional settings or fewer settings may be used according to system requirements.

Referring again to Fig. 16, once one of the capacitor settings is selected in state 1, the state machine moves to state 2 in which the selected capacitor setting is evaluated for bit errors. On entering state 2, a bit error counter and a trial counter are initialized. Bit errors are determined to be those transitions that occur in the forbidden zone. The evaluation in state 2 is intended to provide a fast indication of lock or lack thereof after relatively few trials (only 16 trials lasting approximately 6.6 μs). A trial is an evaluation period on period of measurement described previously lasting for at least the minimum period of the frequency offset.

The risk associated with few trials is that there will be a false lock. The approach described in Fig. 16 uses initially few trials followed by successively larger number of trials. If a false lock occurs, the next series of longer trials should detect

the false lock. Note that the fast evaluation (16 trials) is sensitive to a long string of transitionless bits.

In one embodiment, the control logic is being clocked at a rate significantly slower than the data rate. For example, the data rate may be 2.488320 GHz (OC-48), while the state machine operates at a clock rate of data rate/1024 or 2.43 MHz. For each state machine clock cycle 1024 bits are transmitted and evaluated by the forbidden zone detection circuit. Note that the time period of each evaluation cycle (or trial) corresponds to the 0.4 microseconds measurement period T described above.

If a transition does occur in the forbidden zone during the measurement period T, the bit error counter increments to record the error. The trial counter counts the number of trials over which to evaluate the existence of a bit error during the 1024 bit times. In the illustrated embodiment, 16 evaluation cycles or trials are performed in state 2, which corresponds to approximately 6.6 microseconds. If after 16 evaluation cycles the bit error count is 16, meaning that at least one forbidden zone transition occurred in each of the 16 evaluation cycles, the state machine returns to state 1 to step the digital capacitor settings and thereby adjust the capacitor settings in accordance with, e.g., the stepping algorithm described in Fig. 17. If however, after 16 evaluation cycles are completed, the bit error count is less than 16, then the state machine enters state 3.

State 3 provides a longer period of evaluation than state 2, which helps identify false lock conditions. On entering state 3, the bit error counter and the trial counter, which respectively count the number of detected bit errors and the number of evaluation cycles utilized in the particular state, are both reinitialized, e.g., set to zero. In the illustrated embodiment, 48 cycles are evaluated in state 3. If bit error count is 43 or greater, indicating that lock has not been achieved, the control logic returns to state 1 to step the capacitance settings. In one embodiment, the evaluation of lock in state 3 is robust against 4000 transitionless bits and a maximum bit error rate of $1x10^{-3}$. In the illustrated embodiment, the state 3 evaluation takes approximately 19.6 microseconds given the OC-48 data rates described earlier. If, however at the end of those 48 cycles, the bit error count is less than a predetermined number (43 is the illustrated embodiment), the control logic enters state 4.

State 4 provides a longer period of evaluation than state 3. The bit error counter and the trial counter, which respectively count the number of bit errors and the number of evaluation cycles utilized in the particular state, are both initialized. In the illustrated embodiment, 512 cycles are evaluated in state 4. At the end of those 512 cycles, if the bit error count is less than a predetermined number (497 in the illustrated embodiment), the control logic enters state 5. If however, the bit error count is 497 or greater, indicating that lock has not been achieved, the control logic returns to state 1 to again step the capacitance settings. The evaluation of lock in state 4 is robust against 8000 transitionless bits and a bit error rate of up to $2x10^{-3}$. In the illustrated embodiment, the state 4 evaluation takes approximately 209.7 microseconds, given the rates described earlier. While one preferred embodiment utilizes successively longer testing states, other embodiments may include only one or fewer states. That would entail determining lock after, e.g., 1000 trials.

State 5 is a locked state, and loss of lock (LOL) is deasserted in state 5 to indicate that lock has been achieved. The state machine remains in the locked state evaluating bit errors. The bit errors may again be evaluated over e.g., 512 cycles. At the end of those 512 cycles, if the bit error count is less than a predetermined number (e.g., 497 in the illustrated embodiment), the control logic remains in state 5. If however, the bit error count is 497 or larger, the state machine returns to state one and asserts loss of lock. In the locked state, error monitoring may be selectively enabled. In addition, the capacitor settings for the VCO can be saved (and the accumulator value from the integrating path 240). If lock is lost, those saved values may be used to control the VCO output to output a clock that was recently locked to the input data stream.

With that understanding of how bit errors (transitions in the forbidden zone) are used to determine whether the PLL is locked, an embodiment 1800 of an augmented phase detector that determines when transitions occur in the forbidden zone is illustrated in Fig. 18. The circuit is similar to the phase detector illustrated in Fig. 12, except there are extra latch stages 580, 581 and 1409 and 1411. In the embodiment illustrated in Fig. 18, the latch 540 insulates the earlier signal entering the XOR gate 522 from variations in the timing at node A resulting from varying input data timing (i.e., variations in clock-to-Q timing of register 510 as a function of

its input data setup time). The latches 581 and 1811 ensure that the retimed data is not out of phase with the input data (half a cycle off) and may also be preferable from a loading perspective.

The embodiment illustrated in Fig. 18 also provides a variable length delay, which translates into a phase window appropriate for different data rates. The phase window is a portion of the period of the recovered clock conveyed on node 514. In one embodiment, when the delay path 1815 is selected by selector 1803, the delay is approximately 50-60 picoseconds, which corresponds to an approximately 15% phase window for a 2.5 GHz data rate. However, if a slower data rate is being used, extra delay can be selected by selecting the delay path incorporating extra delay 1801. A rate select signal indicating the data rate frequency may be used for the selector signal 1805. While only two delays are illustrated, multiple delays may be selectable to accommodate a wider range of data rates.

As also illustrated in Fig. 18, when a bit error is indicated on node 1820, the control logic resets the SR latch using the reset signal 1821.

Fig. 19 illustrates another embodiment of an augmented phase detector that determines when transitions occur in the forbidden zone. The circuit is similar to the augmented phase detector illustrated in Fig. 18. The augmented phase detector 1900 includes the phase detector portion 1901 and a bit error detector 1903. The primary difference between the augmented phase detector 1900 and the augmented phase detector 1800 is that the delays are implemented differently. The embodiment illustrated in Fig. 19 is intended to support multiple clock frequencies including the various SONET data rates includes OC–3 (155.520 Mbs), OC–12 (622.080 Mbs), OC-24 (1.244.16 Mbs), and OC–48 (2,488.320 Mbs). There are two delay paths in the bit error detector 1903 based on clock frequency. The clocks for OC24/OC12/OC3 utilize a delay path 1905, while the OC-48 clock utilizes the delay path 1907. A multiplexer 1909 selects the appropriate lower speed clock for delay path 1905. Multiplexer 1911 selects between delay path 1907 and 1905. The data conveyed on node 512 is clocked into register 1913 and latch 1915. The output from the selected delay path is latched into latch 1921. The output from latch 1921 and latch 540 are compared in XOR gate 1923. Note that in a preferred embodiment, all signals shown in Fig. 19 are differential except for rate3, rate48, ber_reset, and

ber_detect. As more fully described herein, the delay in the OC48 path (multiplexer 1911=1) derives its delay as the interpolation between clk48 and phclk as described more fully herein. Note that the phclk conveyed on node 1924 is determined according to the particular clock frequency being supported.

At the beginning of a test cycle (e.g., 1024 data bit times), the ber_reset signal conveyed on node 1930 is set to the opposite binary value of the ber_detect signal conveyed on 1931. At the end of the test interval, if ber_detect equals ber_reset, then an error was not detected by comparator 1923 and register 1933 is not clocked and ber_reset and ber_detect remain at different values. If on the other hand, an error is detected by XOR gate 1923, register 1933 is clocked causing the ber_reset to be clocked into register 1933, converted to a single ended signal in differential to single ended converter 1935, and output on node 1931 as the ber_detect signal.

Fig. 20 illustrates the implementation of the delay associated with the OC3 clock and the generation of the clocks for OC24, OC12 and OC3 mode of operations. All the illustrated signals are differential. The OC-48 clock (clk48) conveyed on node 2001 is divided in half to create clk24 conveyed on node 2003, and further divided to create clk12 conveyed on node 2005.

The phclk3_pre is twice the OC3 (clk3) frequency conveyed on node 2007. The clock phclk3_pre is supplied on node 2009 to multiplexer 1909. The critical edge of phclk3_pre is three half cycles of clk 48 (ideally approximately 600 picoseconds but less in practice due to loading).

Note that, phclk3_pre could also be produced by tapping before the latch 2011 producing clk3. The location of tapping to generate phclk3_pre may depend on loading issues. In the particular implementation shown in Fig. 20, the amplifier 2013 is utilized to minimize the impact of loading. The use of the amplifier lowers the delay between phsclk and phclk3_pre and therefore reduces the forbidden zone. That reduction in the forbidden zone may be undesirable.

Fig. 21 illustrates implementation of the delay associated with OC24 and OC12 clocks. Clk48, clk24, clk12 and clk3 are gated by the respective clock select signals rate48, rate24, rate12 and rate3 signals in buffers 2101, 2103, 2105, and 2107.

Note that all signals shown in Fig. 21 are differential. Phclk_pre12 ideally precedes phclk by half a clk48 period (approximately 200 picoseconds) for OC24 and OC12 clocks. Phclk_pre12 is used for both OC24/OC12. The clock phclk, conveyed on node 1924 to clock the phase detector 1901 is selected by the selector circuit 2100 shown in Fig. 21. For OC48, phclk is delayed from clk48 by the propagation delay of one buffer (approximately 60 picoseconds). However, in some embodiments, a one buffer delay provides a forbidden zone that is too large.

Accordingly, interpolation between the clock phclk conveyed on node 1924 and the clock clk48 conveyed on node 1925 is utilized to achieve a smaller delay and thus a smaller forbidden zone. A smaller forbidden zone gives better jitter tolerance and more accurate ber measurements. In one embodiment, the forbidden zone is approximately 0.1 UI.

Fig. 22 illustrates interpolating between two clocks to achieve a shorter delay than can be achieved by utilizing one buffer. Fig. 22 illustrates a differential latch 2200 which can be cascaded two in a row to form a register with interpolated sample time such as the differential register 1913 (shown in Fig. 19). Assume clk2 is generated by passing clk1 through a differential buffer with a delay of approximately 60 picoseconds. The sample time of the register is determined by the interpolation of the clk1 and clk2 transition edges as set by the relative values of the bias currents Ibias1 and Ibias2. Assume $I_{bias1}+I_{bias2}=I_{bias}$. The sample time is determined by the I_{bias1} and I_{bias2}. For example, if $I_{bias1}=I_{bias}$, and $I_{bias2}=0$, the transition edge of clk1 determines the sample time. If $I_{bias1}=0$, and $I_{bias2}=I_{bias}$, the sample time is determined by the transition edge of clk2. Finally, if $I_{bias1}=I_{bias2}=I_{bias}/2$, the sample time is approximately half way between the transition edges of clk1 and clk2. Thus, the sample time, and thus the delay and the width of the forbidden zone, can be narrowed or lengthened by adjusting the bias currents. Note that the bias current I_{bias} is set according to the voltage drop desired across resistor loads R_L.

Fig. 23 shows a block diagram of an exemplary clock and data recovery circuit that can advantageously exploit the augmented phase detectors and associated control logic described herein (particularly Fig. 19). The clock and data recovery circuit shown in Fig. 23 includes a control circuit 2301 for referenceless frequency acquisition. The control circuit 2301 operates in accordance with the state machine

shown in Fig. 16. The control circuit receives a bit error detect signal conveyed on node 1931 and supplies the reset detector signal on node 1930. As described previously, when the bit error detect signal and reset-detector signal are at different values, a bit error has occurred.

Aside from receiving the bit errors and resetting the detect logic associated therewith, the control circuit also outputs the capacitor settings used by the VCO 260. As described previously, the VCO 260 receives the control signal V_g conveyed on node 252 that adjusts the analog capacitance 410 (Fig. 4). In the illustrated embodiment, the control circuit 2301 supplies the coarse, medium, and fine capacitor control settings for the digitally controlled capacitors 416 (Fig. 4) on node 2310. Node 2310 may be implemented as a multi-bit value supplying, e.g., control values indicating 7 different coarse settings, 8 different medium settings and 6 fine settings for a total of 336 possible capacitor settings.

The control logic also provides a variety of control signals useful in various aspects of the disclosed embodiments. For example, the control logic supplies the mid_rail signal 2304 to the delta-sigma modulator 212 to cause its output to be set to its midpoint when the PLL is operating in a lock to reference mode (state 0). If loss of lock has occurred, asserting the mid_rail signal 2304 zeros the feedforward path 230. The feedforward high bandwidth (ff_high_bw) signal conveyed on node 2306 to the gain block 232 of the feedforward path 230 functions as a bandwidth control signal to select either a higher or lower open loop bandwidth. During acquisition of the frequency of the input data stream, (states 1-4) the feedforward high bandwidth signal is set to maximize the gain to provide a wider bandwidth and thus better lock-in range. The gain is then reduced once confidence of lock is higher or once lock is achieved to provide better performance since a wide lock-in range is no longer needed. The center accumulator (center_accum) signal adjusts the up/down counter in the accumulate block 244 to its center value. When asserted, the integrating path is essentially shut off. An integrating path high bandwidth control signal (int_high_bw) is conveyed on node 2310 to adjust the filter bandwidth provided in filter 248. The center soft switch (center_soft_sw) control signal conveyed on node 2312 to soft switch control circuit 2314 causes the soft switch control to output a center capacitance value during acquisition of the frequency.

In an exemplary embodiment, in state 2 of Fig. 16, center_accum=1, ff_high_bw=1, int_high_bw=1, and mid_rail=0, with a 1 indicating an asserted signal. With control signals at those values, the integrating path is turned off and the feed focused path is set for high gain. In state 3, center_accum=0, ff_high_bw=1, int_high_bw=0, thus enabling the integrating path. In state 4, center_accum=0, ff_high_bw=1, and int_high_bw=0. In the locked state the feedforward high bandwidth signal is deasserted to reduce gain.

Note that the augmented phase detector circuit 1900 in the illustrated embodiment is receiving a clock signal conveyed on node 1924 that in the illustrated embodiment supports SONET OC-48/OC-24/OC12/OC3 data rates (approximately 2.5 GHz for OC-48). The delta-sigma modulator receives a clock at half that rate. The control circuit 2301 receives a clock that is divided down by 1024. One advantage of operating the control circuit 2301 at a much slower clock rate is that it runs slowly enough that it can be easily synthesized by readily available synthesis tools, which results in design savings. In addition, a slower clock rate can save power during operation.

There are two basic errors that can occur utilizing the forbidden zone bit error detection. The first potential error is asserting that the PLL is out-of-lock when it is in fact locked. The probability of that happening depends upon the number of trials that occur as well as the bit error rate of the transmitted data. Referring to Fig. 24A, an exemplary graph illustrates the probability of falsely asserting out-of-lock with various bit error rates (BER) versus various bit error counts for 16 trials. Each of the curves shown in Fig. 24A corresponds to a different bit error rate. The first curve corresponds to a bit error rate of 1×10^{-4}. As the bit error count increases from 4 to 11, the probability of falsely declaring out-of-lock declines from 1×10^{-2} to 1×10^{-9}. That is, the probability declines from one in a hundred to one in a billion. Note that as the bit error rate increases, the probability of falsely declaring out-of-lock also increases. In other words, if there are more bit errors, they are more likely to fall in the forbidden zone resulting in the false declaration of being out-of-lock.

A second potential error that can occur using the forbidden zone approach described herein is to falsely declare the PLL is locked when in fact it is out-of-lock. One parameter that effects the false declaration of lock is the size of the forbidden

zone. If the size of the forbidden zone is increased, more data transitions will occur in the hidden zone and thus more data transitions will be considered bit errors and thus the probability of falsely declaring lock should decrease with a larger forbidden zone. Fig. 24B shows the probability that lock is falsely declared versus count and width of the forbidden zone. When the forbidden zone ($\Delta\Phi/2\pi$) is 10%, with a count of fourteen, the probability of falsely declaring lock is less than 1×10^{-16}. However, for the same count, when the forbidden zone is 5% of the period, the probability of falsely declaring lock is approximately 1×10^{-7}. Thus, the probability of falsely declaring lock decreases with an increasing forbidden zone.

Figs. 24A and 24B shows the probability for 16 trials, which corresponds to state 2 in Fig. 16. The probability curves for state 3 (48 trials) is shown in Figs. 25A and 25B with the probability of falsely declaring out-of-lock shown in Fig. 25A for various bit error rates versus count. As would be expected, the probability of falsely declaring out-of-lock decreases with an increasing count. Thus, with a BER of 1×10^{-3}, the probability of falsely declaring out-of-lock is less than 1×10^{-4} with a bit error count of 42.

Fig. 25B shows the probability of declaring lock falsely for various forbidden zone widths and counts. Fig. 25B shows that the probability of falsely declaring lock is decreased as the forbidden zone increases.

The probability curves for states 4 and 5 (512 trials) are shown in Figs. 26A and 26B. Fig. 26A illustrates the probability of falsely declaring out-of-lock versus count for various bit error rates. As would be expected, the probability of falsely declaring out-of-lock decreases with increasing count. Thus, even with a BER of 2×10^{-3}, the probability of falsely declaring out-of-lock is less than 1×10^{-15} for a bit error count of 500.

Fig. 26B shows the probability of declaring lock falsely for various forbidden zone widths and counts. Fig. 26B shows that the probability of falsely declaring lock decreases with an increased size of the forbidden zone.

The forbidden zone detection circuitry used to determine whether or not the PLL is locked can also be used to indicate the bit error rate once the PLL is locked.

Thus, in the locked state, the state machine can continue to monitor the number of bit errors that occur during, e.g., 512 cycles. If after 512 cycles the number of detected bit errors is less than the predetermined threshold of approximately 500, then the PLL is considered to still be locked, the control logic remains in the locked state, the bit error counter and trial counter are reset, and the monitoring continues for another 512 evaluation cycles.

If the bit error rate is above the threshold count value of approximately 500, the state machine determines the PLL to be out-of-lock and the state machine returns to state 1 and tries to reacquire lock. In trying to reacquire lock, the state machine can enter a hold_vco_state prior to trying to reacquire lock In that state, the accumulator state is maintained (center_accum is set to 0), the center_soft_sw is set to 0, the high bandwidth signals (high_bw) for the feedforward and integrating paths are both set to 0 and mid_rail is asserted. An enable signal for the error monitor function is disabled since the PLL is out-of-lock. The error monitoring function is only valid while the PLL is in-lock. In addition, the control logic delays in this state for 12.3 microseconds to allow for a loss of signal (LOS) exception to occur. That LOS exception can result in LTR being asserted. While in this initialization state, the VCO outputs a clock having a frequency based on the stored VCO settings. Note that the accumulator settings from several intervals ago may be used to control the VCO output.

One advantage of using the forbidden zone detection technique is that in the locked state, if the bit error rate is below the threshold count (e.g., 501), the number of evaluation cycles having transitions in the forbidden zone can be used to indicate the actual bit error rate. In the locked state, the various ranges of bit errors (a bit error being an evaluation cycle having one or more forbidden zone transitions) are assigned a digital value. Figure 27A shows a table of one embodiment of assigning a digital value corresponding to the number of trials (evaluation cycles) having one or more forbidden zone transition for 512 trials in the locked state. That digital value can be stored in a register. In one embodiment, the digital value of the error rate is supplied to a D/A converter that supplies an analog signal, typically an analog current, on an output terminal of the integrated clock and data recovery circuit (see Fig. 45) indicative of the bit error rate. Alternatively, if access is provided to that register, for

instance by a serial communication port, that register can be read to provide an indication of the bit error rate. Note that when the PLL is not locked, the bit error rate indication can be invalidated by setting the output terminal to a maximum or minimum value. Note that if the BER is 1, then a another bit error detection scheme is used to achieve better resolution at low bit error rates.

More specifically, in one embodiment a secondary bit error rate counter is used to provide increased resolution by counting bit errors for a longer time period. More particularly, in one embodiment, the secondary bit error rate counter counts 256 locked-state cycles of 512 trials each. Thus, the secondary bit error rate counter counts for 256 x 512 trials. After each 512 trials, while in the locked state, the lower order bits (e.g., the three low order bits) of the BER counter are added to a secondary BER counter. Fig. 27B illustrates the digital values generated based on the value of the sub bit error count according to one embodiment of the invention.

Referring to Fig. 28, the probability curves for various bit error rates show why it is possible to infer the bit error rate from the count of evaluation cycles having transitions in the forbidden zone. Fig. 28 shows the probability that the measured count of transitions in the forbidden zone (the bit error count) is greater than the bit error rate for 512 trials when the PLL is in-lock. As can be seen, for a bit error rate of 1×10^{-4}, the probability that the bit error count is greater than the bit error rate is negligible if the measured count is greater than approximately 50. For a bit error rate of 1×10^{-3}, the probability that the bit error count is greater than the bit error rate is negligible if the bit error count is greater than approximately 328. Thus, the bit error count determined using the forbidden zone provide a good measure of the bit error rate, even for bit error rates as high as 3×10^{-3} when the bit error count is greater than approximately 489. For low bit error rates additional resolution may be obtained using the sub bit error counter described in relation to Fig. 27B.

In addition to outputting an analog signal indicative of the bit error rate, a BER alarm output terminal may be used to indicate that the bit error rate is above a programmable or fixed threshold value. That threshold value may be determined by an analog signal (e.g., a voltage) supplied to a BER alarm level input terminal of the integrated circuit. The supplied analog voltage may be converted to a digital value and compared with a calculated bit error rate. If the measured bit error rate is above

the BER threshold value, the BER alarm output terminal indicating the bit error rate is above the threshold value is set. In one embodiment the bit error alarm threshold can be set to one of 64 discrete values between error rates of 10^{-3} and 10^{-4} by applying a voltage to the BER alarm level input terminal between 500mV and 2.25V, corresponding to bit error rates of 10^{-3} and 10^{-4}, respectively. That voltage is then converted to the appropriate six bit digital value and compared with the calculated bit error rate to determine whether to assert the BER alarm output terminal. Fig. 41 illustrates the BER alarm output terminal and the BER alarm level input terminal.

The bit error rate, determined as described above, every 512 trials, may be used to control the BER alarm output pin. Alternatively, the bit errors may be counted and the alarm output controlled more frequently if desired. In one embodiment, the BER alarm output is evaluated every approximately 25.6 microseconds. If after 64 trials (at 2.5 MHz), the BER counter is above a critical count value corresponding to the BER alarm level, the BER alarm is asserted. In one embodiment, hysteresis may be selectable so that once the BER alarm is asserted, it is deasserted after 128 or 256 trials. Alternatively, no hysteresis may be utilized. The hysteresis utilized may be programmed over a serial communication port, hardwired in the design or pin programmable.

In addition to monitoring the bit error rate during operation, the bit error rate techniques described herein may also be used advantageously in a manufacturing testing environment. For example in testing the operational capability of a clock and data recovery (CDR) integrated circuit, the bit error rate output signal can be monitored to determine when the CDR integrated circuit has locked onto an input data stream. A high bit error rate indicates a failure to lock whereas a low bit error rate indicates that the CDR has locked to the input data stream. In that way the range of the VCO can be determined by supplying input data streams having different data rates. If a low bit error rate is indicated for a particular data rate, that indicates that the VCO has successfully output a clock signal corresponding to the data rate. Because many CDR parts are intended to work at multiple data rates, the ability to easily test in manufacturing the range of the VCO provides manufactured parts that have been subjected to more rigorous testing and thus parts can be provided with higher quality assurances. One important advantage of using the bit error rate output

signal to determine CDR capability is that while the input data streams may be high speed, the bit error rate signal is a low speed signal easily monitored in a manufacturing and testing environment. Thus, high speed capture and comparison is not necessary for high speed testing.

In addition to testing VCO range, tolerance tests can also be performed using the bit error rate monitoring techniques described herein. In particular, input data streams with various jitter characteristics can be supplied to the CDR circuit and the bit error rate monitored to determine the jitter tolerance of the CDR circuit. As various jitter tolerance requirements can be found in many CDR applications, the ability to readily observe jitter tolerance can also be achieved by monitoring the bit error rate monitor output terminal.

Note that the exact nature of the bit error rate monitoring may vary. For example, an analog bit error signal can be monitored during the testing. If a serial port is available, a bit error rate register can be read out. Alternatively, if an alarm output and threshold is available, the alarm threshold can be adjusted so the alarm is asserted when the bit error rate rises above the threshold value of interest. Thus, various capabilities of the CDR circuit can be evaluated in a manufacturing test environment using bit error rate monitoring.

Referring to Figs. 29-31, exemplary simulations illustrate operation of stepping the capacitance values of the VCO while trying to acquire lock. In the embodiments illustrated in Figs. 29-31, there are assumed to be 7 different coarse capacitance settings and 8 different medium capacitance settings. In addition, fine frequency capacitance settings (6 in an exemplary embodiment) are swept for each of the coarse/medium settings. In the example shown in Figs. 29-31, the top graphs indicate the coarse capacitance settings, the middle graphs indicate the medium capacitance settings and the bottom graphs indicate the cumulative coarse/medium capacitance values. The capacitance values are in femtoFarads. In Fig. 29, the initial medium capacitance setting is 1 and the initial coarse capacitance setting is 1. Note that the stepping of the coarse and medium capacitance settings causes the total capacitance shown in the bottom graph to step successively more positively and negatively around the initial capacitance value determined by an initial coarse and medium setting of one. After 20 different setting combinations, a minimum

capacitance value is reached and the capacitance value is incremented for the remainder of the possible capacitance values. After 56 different settings, the stepping of the capacitance values begins to repeat.

In Fig. 30, the initial medium capacitance setting is 3 and the initial coarse capacitance setting is 4. The stepping of the coarse and medium capacitance settings causes the total capacitance to step successively more positively and negatively around the initial capacitance value determined by the initial medium and coarse setting. After 40 different combinations of course and medium settings, a maximum capacitance value is reached and the capacitance value is decremented for the remainder of the possible capacitance values. After 56 different settings, the stepping of the capacitance values begins to repeat.

In Fig. 31, the initial medium capacitance setting is 7 and the initial coarse capacitance setting is 6, which are maximum values. Since the initial setting is a maximum value, the stepping of the coarse and medium capacitance settings causes the total capacitance to step negatively from the initial maximum capacitance value. After 56 different setting combinations, the minimum capacitance value is reached and stepping of the capacitance values begins to repeat.

While variable capacitance circuits may be particular useful in LC tank circuit implementations of an oscillator, other embodiments may utilize different types of oscillators, e.g., a ring oscillator. In addition, rather than adjusting the capacitance to acquire lock, the voltage or current supplied to a variable oscillator may be adjusted to change frequency until lock is achieved. As long as a control parameter can be adjusted for the particular oscillator used so its output frequency can be varied in response to bit error detection techniques described herein, the type of oscillator that can be used to practice the invention can vary widely and include a wide variety of voltage controlled and current controlled oscillator circuits.

Figs. 32 - 40 illustrate simulations of frequency detection under various scenarios of jitter and transition densities for one embodiment of the invention. The figures show the VCO input and feedforward output (see Fig. 23) in volts. The figures also illustrate accumulator output, the bit error rate counter and the stepping of the fine capacitors settings (0-5). Once the accumulator output begins to become

positive, lock is achieved. In Fig. 32, the transition density is assumed to be 1/2 with no jitter. In Fig. 33, the jitter is 5 Unit Intervals (UI) at 100 kHz. The PLL performance can be seen to exceed the jitter tolerance specifications proposed for SONET/SDH equipment by Bellcore GR-253-CORE, Issue 2, December 1995 and the ITU-T G.958. The x axis shown in Figs. 32 - 40 is in thousands of microseconds.

Fig. 32 shows that the fine capacitor settings were adjusted each time the BER counter reached 16, indicating that lock was not achieved in state 2. After approximately 250 microseconds, the VCO has locked to the input data stream and the fine capacitor settings are no longer being swept, indicating that the coarse and medium settings are stable as well. Note that the accumulator output shown indicates offsets from a center value and not necessarily negative or positive numbers. Note also that the characteristics of the input data stream affect the BER counter. For example, Fig. 36, the BER counter, while locked, indicates a higher bit error rate than Figs. 32-35.

Referring to Fig. 41, the various embodiments described herein are particularly useful in a clock and data recovery (CDR) integrated circuit 370, which is utilized for high speed serial communication systems in which timing information and data is extracted by the PLL 372 from the data supplied on input terminals 371. Note that an analog bit error rate threshold signal, described previously, is supplied on input terminal 373 as the threshold for the bit error rate alarm signal supplied on output terminal 374, which is asserted when the measured bit error rate is above the threshold value. The CDR 370 also supplies a BER output terminal 376, which supplies an analog voltage indicating the bit error rate as previously described herein. Lock to reference (LTR) is an externally provided signal causing the control circuit 2301 (Fig. 23) to remain in state 0. The rate select inputs may be used to select the programmable delay values (see Figs. 18 and 19) in the augmented phase detector utilized in PLL 372.

Thus, various embodiments have been described for locking to a clock embedded in an input data stream without the use of a reference clock. The technique to lock to the clock may also be used to monitor and report the bit error rate and determine if lock is maintained. The description of the invention set forth herein is illustrative, and is not intended to limit the scope of the invention as set forth in the

following claims. Other variations and modifications of the embodiments disclosed herein, may be made based on the description set forth herein, without departing from the scope and spirit of the invention as set forth in the following claims.